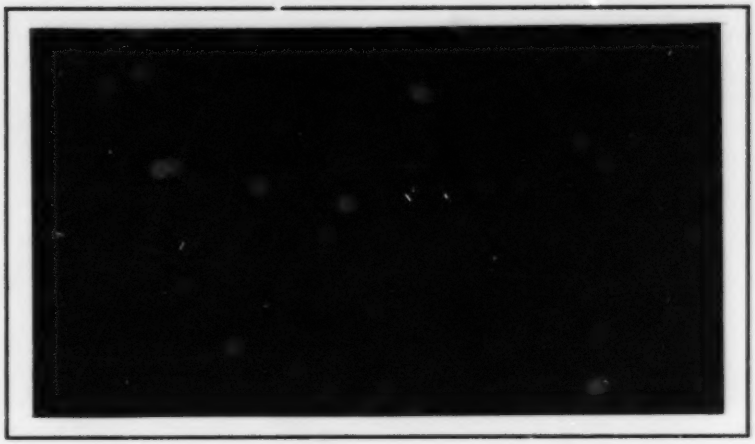


CENTRE FOR
SYSTEMS
SCIENCE

LCCR

Simon Fraser University, Burnaby, B.C., Canada V5A 1S6 / (604) 291-4704

Bounds on the number of hidden units of Boltzmann Machines

Eduard Moser and Tiko Kameda

CSS/LCCR TR90-19

November 1990

92-05801/

Bounds on the number of hidden units of Boltzmann machines*

Eduard Moser † and Tiko Kameda ‡

Abstract

It is known that any given probability distribution of the states of the observable units of a Boltzmann machine can be realized if no limit is imposed on the number of hidden units [AaK89]. But very little is known about the number of hidden units necessary for such realization. We consider Boltzmann machines as associative memories and show that there exist vector sets whose memorization on a Boltzmann machine requires a number of hidden units exponential in the size of the vectors (i.e., the number of components in each vector). Additional results give tight bounds on the number of hidden units in terms of the vector set size (i.e., the number of vectors in the set). Furthermore, our restrictive model of auto-association in Boltzmann machines provides a construction for negation, intersection, and composition of the vector sets which are memorized by Boltzmann machines.

---

* This work was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada and the Advanced System Institute of British Columbia.

† Universität Karlsruhe, Informatik, D-7500 Karlsruhe, eduard@cs.sfu.ca

‡ School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6

1. Introduction

A neural network can be represented by a graph whose vertices are called *units* and whose edges *connections*. Each neural network has a specific transition function, which determines the *activity* of the unit based on its inputs. In our model, the activity of a unit is also considered as its output. The *connection strength* associated with a connection determines the mutual influence of the two units connected by it. The power of neural networks comes from the large number of highly interconnected units.

The Boltzmann machine [AHS85] is one such neural network, which is characterized by its probabilistic transition function. Depending on a global parameter, called *temperature*, the transition function stochastically determines the binary activity value of a unit based on its inputs. This transition function is such that the probability distribution of the *system states* (the vectors formed by the activity values of all units) follows the Boltzmann distribution. If we want such a machine to have a particular state probability distribution, we have to find suitable connection strengths. This task is usually accomplished by a learning algorithm.

In general, we do not observe the whole system but a subset of units, called the *observable units*. The other units are called the *hidden units*. The vector formed by the activity values of the observable units is called an *observable state*. The learning task is to find suitable connection strengths, such that the probability distribution of the observable states meets our requirements. One such problem is to find a machine which "computes" a given boolean function f in the following sense. One of the observable units is designated as the system output and the others function as inputs providing boolean values. Let $\hat{x}$ stand for a vector of these boolean values. A Boltzmann machine is said to realize a boolean function $f(\hat{x})$ if for all inputs $\hat{x}$, the probability of the observable state $(\hat{x}, f(\hat{x}))$ is higher than the probability of the state $(\hat{x}, \overline{f(\hat{x})})$.

Boltzmann machines without hidden units are thoroughly discussed in [AaK89]. They show that the learning algorithm will always find a solution, if one exists. However, only certain probability distributions can be reached. For instance, the boolean function exclusive-or cannot be realized without hidden units. If any number of hidden units can be used, any probability distribution can be reached (with error smaller than any given $\epsilon > 0$ for the probability of each observable state). But very little is known as to how many units are necessary. Even with a network of the structure of a bipartite graph with the observable units forming one side of the graph, all probability distributions can be realized. It is not known, however, if any further extension of this structure will lead to fewer hidden units. This paper deals mainly with this simplest structure with hidden units, but, as mentioned later, the results we will derive also hold for all structures without connections among observable units. Using Boltzmann machines of the structure of bipartite graphs, we can synchronously simulate the machines by alternately calculating each side in parallel. This simulation will approach the same probability distribution of its states as asynchronous simulation.

In the rest of this section, we give all necessary definitions and introduce the concept of auto-associative set as the set of vectors which is "memorized" by a Boltzmann machine.

In section 2, some results about Boltzmann machines are derived, including constructions for negation, intersection, and composition, as well the fact that any Boltzmann machine can be described in space polynomial in the number of its units. This gives a lower bound on the number of the hidden units in terms of the size of the vectors in the auto-associative set. In section 3, we consider the so-called structured Boltzmann machines and derive some more bounds on the number of hidden units in terms of the size of the auto-associative set.

1.1 Boltzmann machine

A Boltzmann machine is defined by its units $\{u_1, \ldots, u_m\}$, its connection strengths $\{w_{ij} \mid i, j = 1, \ldots, m\}$ between pairs of units, and a threshold value Θ_i for each unit u_i. The connection strengths and the thresholds are rational numbers. Further, the connection strengths are symmetric $w_{ij} = w_{ji}$ and $w_{ii} = 0$. The *system state* is denoted by $\hat{s} = (s_1, \ldots, s_m) \in \{0,1\}^m$, where s_i is the activity on the unit u_i. The *energy* of system state $\hat{s}$ is defined by

$$E(\hat{s}) = -\sum_{i<j} w_{ij} s_i s_j + \sum_i \Theta_i s_i.$$

Let $P(\hat{s})$ denote the probability of system state $\hat{s}$. The probability distribution of the system states follows the Boltzmann distribution

$$\frac{P(\hat{s}_1)}{P(\hat{s}_2)} = exp(-(E(\hat{s}_1) - E(\hat{s}_2))/c), \tag{1}$$

where c is a positive constant, which we call *temperature*. This probability distribution is reached at *thermal equilibrium*, after infinite time regardless of the initial distribution.

A Boltzmann machine can be simulated by Monte Carlo method [BiY86]. At each simulation step one unit is arbitrarily chosen and its activity value calculated depending on its neighbors as follows. The probability that the activity of a unit is set to 1 can be derived from (1):

$$P(s_i = 1) = \frac{1}{1 + exp(-sum_i/c)},$$
$$\text{where} \quad sum_i = \sum_{i \neq j} w_{ij} s_j - \Theta_i.$$

After certain time of calculating, depending on energy levels and temperature, the probability distribution satisfying (1) will be reached within some small error. In analogy to statistical mechanics, the probability distribution approaches the uniform distribution as temperature increases.

1.2 Auto-associative set

Our view of auto-associative memory is a system which "stores" a set of vectors. An inquiry is made by presenting any vector or part of a vector, and the system will respond with a vector from the set, which is the "most closely related." The vector set is a subset of the n-dimensional binary vector space $\{0,1\}^n$. Auto-associative memory can be considered as associative memory if we store each of the associative pairs as one vector. If one part of a pair is presented, the system will provide the other part.

To use a Boltzmann machine as auto-associative memory we divide its set of units into observable and hidden units. The system state $\hat{s}$ can be written as $\hat{s} = (\hat{\alpha}, \hat{\beta})$ such that $\hat{\alpha}$ represents the activity on the observable units and $\hat{\beta}$ represents the activity on the hidden units. Only the observable units can be seen by the user. The probability of observable state $\hat{\alpha}$ can be expressed as a function of temperature c

$$P(\hat{\alpha}, c) = \sum_{\hat{\beta}} P((\hat{\alpha}, \hat{\beta}), c).$$

In the following, we always assume that the observable states are members of the vector space $\{0,1\}^n$.

A Boltzmann machine *BM stores* a vector set VS if the probability distribution of its observable states satisfies:

$$\exists\, c_0 \quad \forall\, c < c_0 \quad \forall\, \hat{\alpha} \in VS \quad \forall\, \hat{\alpha}' \notin VS \;:\quad P(\hat{\alpha}, c) > P(\hat{\alpha}', c), \tag{2}$$

where c_0 and c denote temperatures.

The gap between the probabilities of the members of set VS and the vectors which are not in VS can be enlarged arbitrarily by lowering temperature. If VS is stored by BM then

$$\forall\, d > 0 \quad \exists\, c_0 \quad \forall\, c < c_0 \quad \forall\, \hat{\alpha} \in VS \quad \forall\, \hat{\alpha}' \notin VS \;:\quad P(\hat{\alpha}, c) > d \cdot P(\hat{\alpha}', c). \tag{3}$$

The Boltzmann machine can be applied for language recognition [PaS89]. In this application, we use only one observable unit for the system output and the other observable units are used for word input. Similar to language recognition is the realization of boolean functions on a Boltzmann machine. The vector set for a function $f : \{0,1\}^k \rightarrow \{0,1\}$ can be described as

$$VS = \{(\hat{x}, f(\hat{x})) \,|\, \hat{x} \in \{0,1\}^k\}.$$

A Boltzmann machine implementing this VS has $k+1$ observable units, one of which (u_{k+1}) carries the system output.

The probability of realizing a boolean function correctly can be derived from the probability distribution of its observable states. Let $P(f(\hat{x}) \,|\, \hat{x})$ denote the probability that $f(\hat{x})$ is the system output for a given input $\hat{x}$, where

$$P(f(\hat{x}) \,|\, \hat{x}) = \frac{P(\hat{x}, f(\hat{x}))}{P(\hat{x}, f(\hat{x})) + P(\hat{x}, \overline{f(\hat{x})})}. \tag{4}$$

[PaS89] uses a certain probability p_0 to define correct realization by:

$$\forall\, \hat{x} \quad P(f(\hat{x}) \,|\, \hat{x}) > p_0, \qquad (\text{or} \quad P(\overline{f(\hat{x})} \,|\, \hat{x}) \leq 1 - p_0). \tag{5}$$

If $\{\hat{x}, f(\hat{x})\}$ is realized by *BM* with probability $\frac{1}{2}$ then the machine can fulfill (5) for any $p_0 < 1$ by choosing $d = p_0/(1 - p_0)$ of (3) and some temperature.

Not only can the gap between the probabilities of members and non-members of the vector set be enlarged, but the gap between the probabilities of the members enlarges arbitrarily as temperature decreases. For $\{\hat{\alpha}_1, \hat{\alpha}_2\} \subseteq VS$ which are stored in *BM*, if $min_{\hat{\beta}}(E(\hat{\alpha}_1, \hat{\beta})) < min_{\hat{\beta}}(E(\hat{\alpha}_2, \hat{\beta}))$ holds, then we have

$$\forall\, d > 0 \quad \exists\, c_0 \quad \forall\, c < c_0 \;:\quad P(\hat{\alpha}_1, c) > d \cdot P(\hat{\alpha}_2, c)$$

or

$$\forall\, \epsilon > 0 \quad \exists\, c_0 \quad \forall\, c < c_0 \;:\quad P(\hat{\alpha}_2, c) < \epsilon.$$

Real simulations last only finite time and never reach thermal equilibrium, but a so called "quasi" equilibrium at which the probability of each state is within some error close to the probability of (1). To satisfy (5) even for elements of the vector set with very small probability, as shown above, we have to run the machine for a very long time to achieve an error small to these small probabilities.

To avoid unacceptable calculation times, we have to avoid these uneven probabilities. For that reason, we introduce a more convenient but more restrictive definition of what a Boltzmann machine memorizes. Considering energy, we can derive from (3) the following fact. If a Boltzmann machine stores a vector set *VS*, then there exists a *critical energy* E_{crt} for which the following hold:

$$\begin{aligned} \forall\, \hat{\alpha} \in VS \quad \exists\, \hat{\beta} \;&:\quad E(\hat{\alpha}, \hat{\beta}) \leq E_{crt}, \\ \forall\, \hat{\alpha}' \notin VS \quad \forall\, \hat{\beta}' \;&:\quad E(\hat{\alpha}', \hat{\beta}') > E_{crt}. \end{aligned}$$

If we choose the critical energy equal to the *minimal energy* $E_{min} = min_{\hat{s}}(E(\hat{s}))$, then Boltzmann machine *BM* stores one particular vector set, called the *auto-associative set* of *BM*. More formally, the auto-associative set *AS* of a Boltzmann machine *BM* is defined by:

$$AS(BM) = \{\hat{\alpha} \,|\, \exists\, \hat{\beta}\; E(\hat{\alpha}, \hat{\beta}) = E_{min}\}.$$

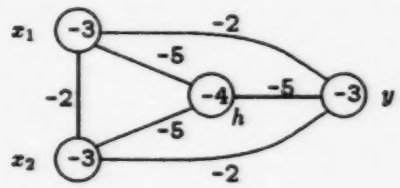
The probability distribution of the auto-associative set satisfies

$$\exists\, c_0, d \quad \forall\, c < c_0 \quad \forall\, \{\hat{\alpha}, \hat{\alpha}'\} \subseteq AS(BM) \;:\quad P(\hat{\alpha}', c) < d \cdot P(\hat{\alpha}, c),$$

which implies that $\exists\, \epsilon\, \forall\, c \;:\quad min_{\hat{\alpha} \in VS}(P(\hat{\alpha}, c)) > \epsilon > 0.$

Figure 1 shows a Boltzmann machine implementing the exclusive-or function on two inputs x_1 and x_2. x_1, x_2, and y are the observable units and h is the hidden unit. Each node is labeled by its threshold and each connection is labeled by its connection strength. The table lists the energy and probability (with temperature $c = 0.5$) for each state. The states with the lowest energy (-4) define the auto-associative set, which is:

$$AS(BM_{xor}) = \{(x_1, x_2, x_1 \oplus x_2)\} = \{(0,0,0), (0,1,1), (1,0,1), (1,1,0)\}.$$



(x_1,x_2,y,h)	Energy	Probability
(1,1,0,0)	-4	0.21755
(1,0,1,0)	-4	0.21755
(0,1,1,0)	-4	0.21755
(0,0,0,1)	-4	0.21755
(1,0,0,0)	-3	0.02944
(0,1,0,0)	-3	0.02944
(0,0,1,0)	-3	0.02944
(1,1,1,0)	-3	0.02944
(1,0,0,1)	-2	0.00398
(0,1,0,1)	-2	0.00398
(0,0,1,1)	-2	0.00398
(0,0,0,0)	0	0.00007
(1,1,0,1)	2	0.00000
(1,0,1,1)	2	0.00000
(0,1,1,1)	2	0.00000
(1,1,1,1)	8	0.00000

Fig. 1: Exclusive-or and its energy table.

2. Some general results

2.1 Negation

In this section we show how to construct from a given Boltzmann machine *BM* with auto-associative set $AS(BM)$ another Boltzmann machine BM' with auto-associative set $AS(BM') = \{\hat{\alpha} \oplus \hat{\gamma} \mid \hat{\alpha} \in AS(BM)\}$ for any given $\hat{\gamma} \in \{0,1\}^n$. This construction can be used to realize the complement of a boolean function. If $AS(BM) = \{(\hat{x}, f(\hat{x})\} \subseteq \{0,1\}^n$ and $\hat{\gamma} = \hat{e}_n$ then the auto-associative set of BM' is $\{(\hat{x}, \overline{f(\hat{x})}\}$. Lemma 1 below gives a more general scheme to invert any activity value in a Boltzmann machine.

Lemma 1: For any Boltzmann machine *BM* there exits a machine BM' such that if $\hat{s} = (s_1, \ldots, s_i, \ldots, s_m)$ is a system state of *BM* then $\hat{s}' = (s_1, \ldots, \overline{s_i}, \ldots, s_m)$ is a state of BM' and

$$E_{BM}(\hat{s}) = E_{BM'}(\hat{s}') + C, \qquad \text{where } C \text{ is a constant.}$$

Proof: Assume without loss of generality that $i = 1$. Then the energy of *BM* can be written as:

$$\begin{aligned} E_{BM}(\hat{s}) &= -\sum_{1<j} w_{1j}s_1s_j + \Theta_1 s_1 - \sum_{1<i<j} w_{ij}s_is_j + \sum_{1<i} \Theta_i s_i \\ &= -\sum_{1<j} w_{1j}(1-\overline{s_1})s_j + \Theta_1(1-\overline{s_1}) - \sum_{1<i<j} w_{ij}s_is_j + \sum_{1<i} \Theta_i s_i \\ &= +\sum_{1<j} w_{1j}\overline{s_1}s_j - \Theta_1\overline{s_1} - \sum_{1<i<j} w_{ij}s_is_j + \sum_{1<i} (\Theta_i - w_{1i})s_i + \Theta_1 \end{aligned}$$

Let us define BM' to be the same as *BM* except that:

$$\Theta'_1 = -\Theta_1,$$
$$\Theta'_i = \Theta_i - w_{1i} \quad \text{and} \quad w'_{1i} = -w_{1i} \qquad \forall\, i,\ 1 < i \leq n.$$

Then BM and BM' satisfy the equality in the lemma with $C = \Theta_1$. ♡

Lemma 1 shows that for each machine there exists a corresponding machine with a similar energy landscape over its system states, but with the negation of the activity of one unit. Since the difference in energy between the two machines is constant for all system states, they have the same probability distribution.

By applying this lemma we can derive the following corollary:

Corollary 1: Let $\hat{\gamma} \in \{0,1\}^n$. For any Boltzmann machine BM there exists a Boltzmann machine BM' with the same number of units such that $AS(BM') = \{\hat{\alpha} \oplus \hat{\gamma} \mid \hat{\alpha} \in AS(BM)\}$. ♡

Figure 2 shows a Boltzmann machine $BM_{\overline{xor}}$ which realizes the complement of exclusive-or. Its auto-associative set is:

$$AS(BM_{\overline{xor}}) = \{(x_1, x_2, \overline{x_1 \oplus x_2})\} = \{(0,0,1),(0,1,0),(1,0,0),(1,1,1)\}.$$

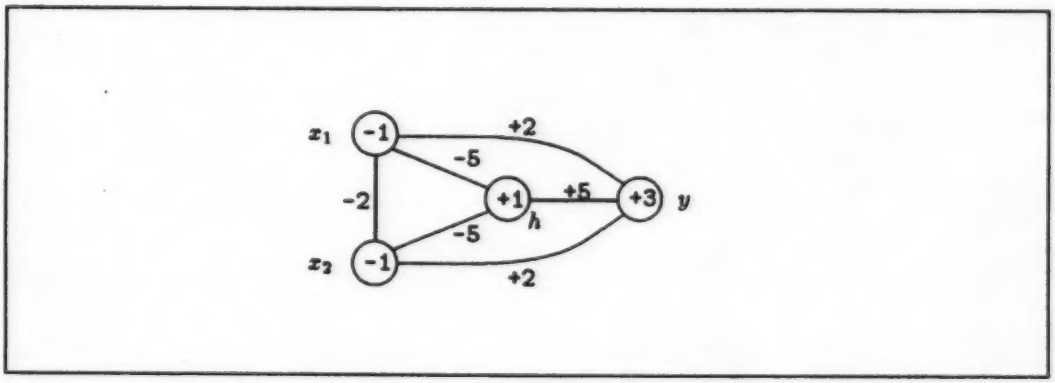


Fig. 2: Complement of exclusive-or.

2.2 Intersection

If the intersection of the auto-associative sets of two Boltzmann machines BM_1 and BM_2 is not empty then we can construct BM_3 with auto-associative set $AS(BM_3) = AS(BM_1) \cap AS(BM_2)$. The construction simply merges the observable units of BM_1 and BM_2. The hidden units and the connections remain the same. The thresholds of the new observable units become the sum of the thresholds of the original units. The machine BM_3 can be specified by the following formal description:

For i=1,2, and 3, let $\{u_1^i, \ldots, u_{n+l_i}^i\}$ be the units of BM_i, first n of which are observable units, let $\{w_{jk}^i\}$ be its connection strengths, and let $\{\Theta_j^i\}$ be its thresholds. We define BM_3 by:

$$\Theta_j^3 = \begin{cases} \Theta_j^1 + \Theta_j^2 & 1 \le j \le n, \\ \Theta_j^1 & n < j \le n + l_1, \\ \Theta_{j-l_1}^2 & n + l_1 < j \le n + l_1 + l_2, \end{cases}$$

$$w_{jk}^3 = \begin{cases} w_{jk}^1 + w_{jk}^2 & 1 \le j, k \le n, \\ w_{jk}^1 & 1 \le j \le n + l_1,\ n < k \le n + l_1, \\ w_{j,k-l_1}^2 & 1 \le j \le n,\ n + l_1 < k \le n + l_1 + l_2 \\ w_{j-l_1,k-l_1}^2 & n + l_1 < j, k \le n + l_1 + l_2, \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 2: If $AS(BM_1) \cap AS(BM_2) \neq \emptyset$ for Boltzmann machines BM_1 and BM_2 then the construction above gives BM_3 with auto-associative set $AS(BM_3) = AS(BM_1) \cap AS(BM_2)$.

Proof: We defined BM_3 in such a way that its energy is the sum of the energies of BM_1 and BM_2. To be more specific, let $(\hat{\alpha}, \hat{\beta}_1)$ be a system state of BM_1, $(\hat{\alpha}, \hat{\beta}_2)$ a system state of BM_2, and E_i the energy of BM_i. Then

$$E_3(\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2) = E_1(\hat{\alpha}, \hat{\beta}_1) + E_2(\hat{\alpha}, \hat{\beta}_2).$$

Therefore, the minimal energy of BM_3 satisfies $E_{min}^3 \ge E_{min}^1 + E_{min}^2$. Since there exists $\hat{\alpha} \in AS(BM_1) \cap AS(BM_2)$, we have

$$E_{min}^3 = E_{min}^1 + E_{min}^2.$$

Hence it follows that $AS(BM_3) = AS(BM_1) \cap AS(BM_2)$. ♡

2.3 Composition

In many applications, composition is a way to build complex structures by using simple structures. Here, composition refers to functions implemented by the auto-associative sets of Boltzmann machines. Consider the following functions $f_1, \ldots, f_m, g$:

$$f_i : \hat{x} \rightarrow f_i(\hat{x}), \quad \hat{x} \in \{0,1\}^k, \quad f_i(\hat{x}) \in \{0,1\}, \quad 1 \leq i \leq m,$$
$$g : \hat{y} \rightarrow g(\hat{y}), \quad \hat{y} \in \{0,1\}^m, \quad g(\hat{y}) \in \{0,1\}.$$

We define a composition of $f_1, \ldots, f_m, g$ to be the function h:

$$h : \hat{x} \rightarrow g(f_1(\hat{x}), \ldots, f_m(\hat{x})).$$

Using Boltzmann machines which realize all f_i and g as building blocks, we can construct a machine which realizes h. The construction is carried out in three steps:

First, we define a new set of Boltzmann machines whose auto-associative sets are subsets of $\{0,1\}^{k+m+1}$. The auto-associative set of BM'_i, $1 \leq i \leq m$, is given by

$$AS(BM'_i) = \{\hat{x}, y_1, \ldots, y_{i-1}, f_i(\hat{x}), y_{i+1}, \ldots, y_{m+1} \mid \hat{x} \in \{0,1\}^k, \ y_j \in \{0,1\}\}$$

and that of $BM'_{,n+1}$ is given by

$$AS(BM'_{m+1}) = \{\hat{x}, \hat{y}, g(\hat{y}) \mid \hat{x} \in \{0,1\}^k, \ \hat{y} \in \{0,1\}^m\}.$$

For each i, BM'_i is a copy of BM_i with some additional observable units. The additional units have no connections and their thresholds are all zero (dummies).

Second, we use lemma 2 to construct a machine which realizes the intersection of all $m+1$ auto-associative sets. We construct a machine BM'_{comp} with

$$AS(BM'_{comp}) = \{\hat{x}, f_1(\hat{x}), \ldots, f_m(\hat{x}), g(f_1(\hat{x}), \ldots, f_m(\hat{x})) \mid \hat{x} \in \{0,1\}^k\}.$$

Finally, we design BM_{comp} as a copy of BM'_{comp} but the observable units of BM'_{comp} which represent $f_1(\hat{x}), \ldots, f_m(\hat{x})$ are now hidden units.

Lemma 3: Let $f_i : \{0,1\}^k \rightarrow \{0,1\}$, $1 \leq i \leq m$, and $g : \{0,1\}^m \rightarrow \{0,1\}$ be binary functions, and let $VS_i = \{(\hat{x}, f_i(\hat{x}))\}$, $i \leq m$, and $VS_{m+1} = \{(\hat{y}, g(\hat{y}))\}$. For each i, $1 \leq i \leq m+1$, let BM_i be a Boltzmann machine such that $AS(BM_i) = VS_i$. The construction above gives BM_{comp} with auto-associative set

$$AS(BM_{comp}) = \{(\hat{x}, g(f_1(\hat{x}), \ldots, f_m(\hat{x}))\},$$

the number of whose hidden units is at most the sum of the numbers of hidden units of all BM_i, $1 \leq i \leq m+1$.

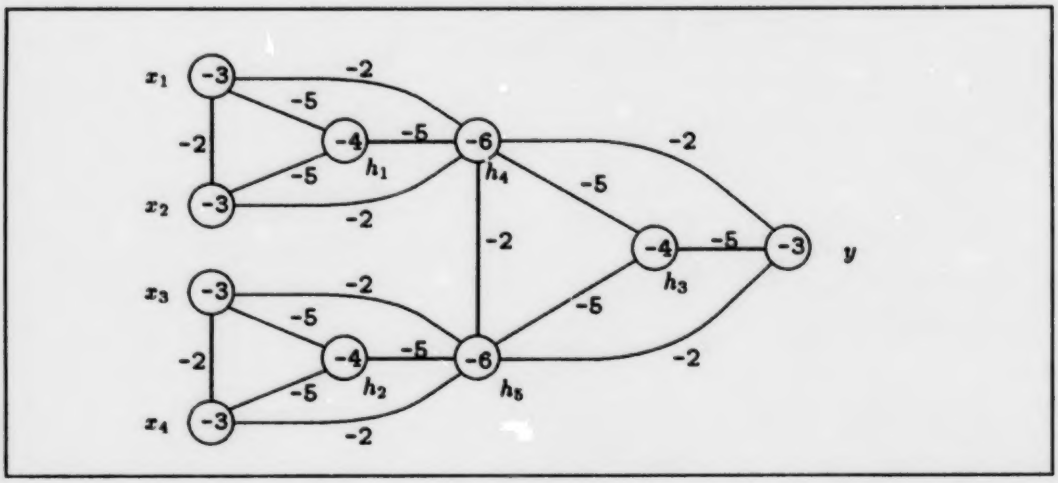


Fig. 3: Composition of three exclusive-or Boltzmann machines ($k = 4, m = 2$).

Proof: The first and third steps do not affect the behavior of any machine. For the second step, we have to show that lemma 2 is applicable. The intersection of all auto-associative sets is not empty. Therefore, lemma 2 is applicable and the second step can be processed. ♡

The same construction can be used even if the functions $\{f_i\}$ are defined over different domains. We add dummies to each BM_i such that each realized function is defined over one common input domain. For example, let $h(x, y) = g(f_1(x), f_2(y))$, and suppose that BM_1 realizes $f_1(x)$ and BM_2 realizes $f_2(y)$. We form new machines realizing $f_1'(x, y) = f(x)$ and $f_2'(x, y)$ by adding dummy units to BM_1 and BM_2, respectively. Now we can apply the construction and obtain a Boltzmann machine realizing $h(x, y)$.

Figure 3 shows the composition of three exclusive-or machines, which realizes the parity function for four inputs. Its auto-associative set is:

$$AS = \{(x_1, x_2, x_3, x_4, x_1 \oplus x_2 \oplus x_3 \oplus x_4)\}.$$

Each of the Boltzmann machines realizing $x_1 \oplus x_2$ and $x_3 \oplus x_4$ is extended by two dummy units, such that they are defined over $\{0, 1\}^4$.

Remark: The given construction for merging observable units can be extended such that any observable unit of BM_{comp} corresponds to a set of observable units of BM_i and not only in the hierarchical manner. In the case stated in lemma 3, the intersection of all given auto-associative sets is never empty. An example in which the intersection is empty is shown in Figure 4. Here we composed three machines in a cycle, each realizing negation

$$AS_{neg} = \{(\hat{x}, \overline{\hat{x}})\} = \{(0, 1), (1, 0)\}.$$

The intersection is empty and $E_{min}^{comp} > 3 \cdot E_{min}^{neg}$; therefore, the above construction does not work.

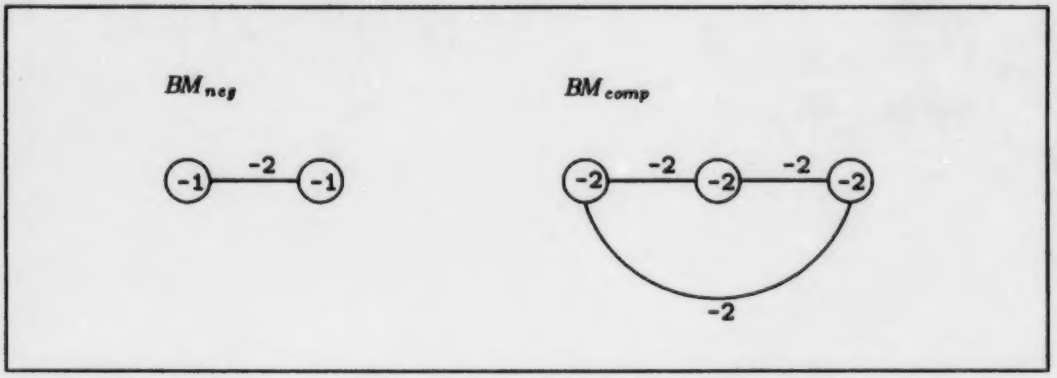


Fig. 4: Composition which violates the composition rule.

Computational power of Boltzmann machines

With composition (lemma 3) and one simple Boltzmann machine which implements the NAND function, we can compare the computational power of Boltzmann machines and switching networks.

Theorem 1: Boltzmann machines have at least as great computational power as switching networks.

Proof: Let Boltzmann machine BM_{nand} consists of three units $\{u_1, u_2, u_3\}$ and

$$w_{1,2} = -1 \quad w_{1,3} = w_{2,3} = -2 \quad \Theta_1 = \Theta_2 = -2 \quad \Theta_3 = -3.$$

Composing BM_{nand}, which realizes the NAND function, arbitrary switching networks can be realized. ♡

Remark: This result says nothing about the computation time needed to simulate a switching network. The *computational time* of a Boltzmann machine can be defined as the time needed to reach a certain probability p_0 for which the result is the desired output (as stated in equation (5)). Assume that computing the negation of one input needs time t_s on a switching network and time t_b on a Boltzmann machine. A chain of n inverters a switching network needs time $n \cdot t_s$. The Boltzmann machine constructed by lemma 3 needs at least time $n^2 t_b$.

2.4 Bounds on the number of hidden units

The notion of auto-associative set gives us the possibility to estimate the amount of information coded in a Boltzmann machine. For this estimation we do not want to distinguish between Boltzmann machines with the same auto-associative set. We say that two Boltzmann machines are *equivalent*, iff their auto-associative sets are identical, otherwise the machines are *distinct*. The following theorem gives a polynominal bound on the amount of space necessary to describe an arbitrary Boltzmann machine. The proof finds for any machine an equivalent version whose connection strengths and thresholds are all integers of bounded size.

In [AbJ85] some stronger results are derived for Hopfield nets. But these results are not applicable on Boltzmann machines. It is easy to construct machines which are "non-distinguishable" in the sense of that paper, but realize different functions (and which are, of course, distinct in the sense of this paper).

Making use of the space bound, we can derive the number of distinct machines with the same number of units. By comparing the number of distinct machines with the number of different subsets of $\{0,1\}^n$, we finally obtain a bound on the number of hidden units necessary to code an arbitrary subset of $\{0,1\}^n$.

Theorem 2: For any Boltzmann machine with n observable and l hidden units, there exists an equivalent version, which can be described with less than $(n+l)^5$ bits.

Proof: Let BM be a Boltzmann machine with n observable and l hidden units. Without loss of generality we assume $\hat{0} \in AS(BM)$ and $E(\hat{0},\hat{0}) = E_{min}$ (see Lemma 1). Then $E_{min} = 0$ holds. The system state $\hat{s} = (\hat{\alpha},\hat{\beta}) = (s_1,\ldots,s_{n+l})$. Further, we assume that $min\{E(\hat{\alpha},\hat{\beta}) \mid E(\hat{\alpha},\hat{\beta}) \neq E_{min}\} = 1$. The energy defines an equation system:

$$E(\hat{\alpha}_i,\hat{\beta}_i) \begin{cases} = 0. \\ \geq 1. \end{cases}$$

Suppose thresholds and connection strengths are variables. Then any solution of the equation system would give a Boltzmann machine which is equivalent to BM. A solution in which the variables take the smallest values is bounded by at most $m = \frac{1}{2}((n+l)^2 + n + l)$ constraints with right-hand sides equal to 0 or 1. We represent these m constraints in matrix form:

$$A\hat{r} = \hat{b}.$$

$\hat{r}$ contains the variables, the thresholds ($r_j = \Theta_j$) and connection strengths ($r_k = -w_{ij}$). A contains coefficients which have the form s_j^i, or $s_j^i s_k^i$. $\hat{b}$ contains the right-hand side of the equations. Without loss of generality, we assume $Det(A) \neq 0$. Then $\hat{r}$ can be solved:

$$\hat{r} = A^{-1}\hat{b}.$$

The components of $\hat{r}$ have the form

$$r_i = \frac{1}{Det(A)} \sum_{j=1}^{m} (-1)^{i+j}\, Det(A_{ij})\, b_j.$$

A_{ij} denotes a minor of A. $\hat{r}^* = |Det(A)| \cdot \hat{r}$ defines another Boltzmann machine BM' with auto-associative set $AS(BM)$, whose thresholds and connection strengths are all integers. Since all coefficients in A and $\hat{b}$ are 0 or 1, all $Det(A_{ij})$ are bounded by $m!$ and

$$\forall\, i: \quad r_i^* \leq m \cdot m! \leq 2^{m\, log(m)}.$$

Therefore, the complete description of the Boltzmann machine can be done within

$$m^2\, log(m) < (n+l+1)^4\, log(n+l+1) < (n+l)^5 \quad \text{bits.} \qquad \heartsuit$$

The following bound on the number of distinct Boltzmann machines follows directly from the space bound.

Corollary 2: There are less than $2^{(n+l)^5}$ distinct Boltzmann machines with n observable units and l hidden units. $\heartsuit$

Now, we can derive a bound on the number of hidden units considering all possible vector sets of constant size; the bound is given in terms of the size of the vectors (i.e., number of components).

Corollary 3: There exists a vector set $VS \subseteq \{0,1\}^n$ whose implementation on a Boltzmann machine needs at least $2^{\frac{n-1}{5}}$ hidden units.

Proof: There exist 2^{2^n} subsets of $\{0,1\}^n$. To implement all of them, we need at least the same number of distinct Boltzmann machines. Let n be the number of observable units and l be the number of hidden units. Then we can derive:

$$2^{(n+l)^5} \geq 2^{2^n} \qquad \Longrightarrow \qquad l > 2^{\frac{n-1}{5}}. \qquad \heartsuit$$

Remark: The bounds stated in Corollary 2 and Corollary 3 can still be improved by counting machines which are equivalent but have a different representation on their hidden units.

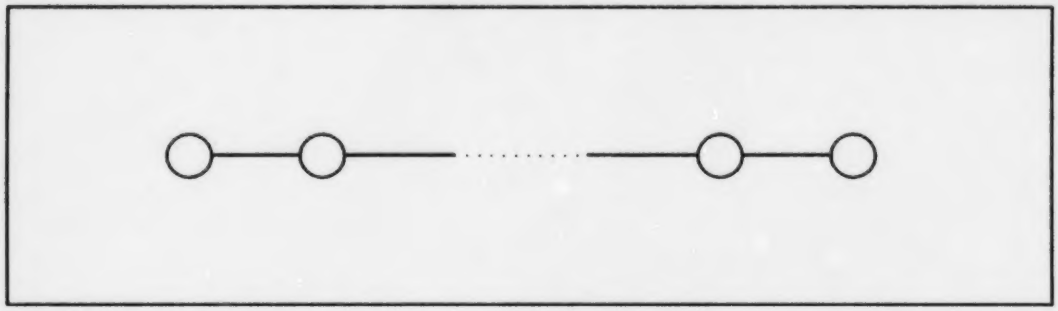


Fig. 5: Chain of k units.

3. Structured Boltzmann machines

As a remark in section 2.3 we mention the computational time in chains. But even worse than that is the behavior during *learning*. Learning transforms a given Boltzmann machine by changing its connection strengths and its thresholds such that the new machine better approximates a given probability distribution of its observable states. The learning algorithm in [AHS85] performs this task. But only a few types of Boltzmann machines are appropriate for this learning strategy.

Let us consider a chain of k units (Figure 5). The first and the last units are the observable units and the machine is supposed to have the same activity value on both ends of the chain. It can be shown that the learning algorithm will approximate the goal for any k. But a numerical analysis proves that the time needed for learning this identity function on the chain is exponential in k [Mos90].

As discussed in [AaK89], structures with hidden units can get stuck in local minima during learning. There is evidence that connections between hidden units may cause the algorithm to fail

Both deficiencies, the long learning time and the possible failure of a learning process, motivates our further investigations into certain structured Boltzmann machines.

Boltzmann machines can be classified by their underlying structures. A structured Boltzmann machine doesn't contain all possible connections. If a connection between two units doesn't exist then the connection strength is considered to be zero.

A 2-layer structured Boltzmann machine (2-LSBM) consists of n observable units $\{u_1^o, \ldots, u_n^o\}$, l hidden units $\{u_1^h, \ldots, u_l^h\}$, and $n * l$ connections with connection strength matrix $W = \{w_{ij}\}$, where w_{ij} is the connection strength between u_i^o and u_j^h. Almost all examples for learning in [AaK89,AHS85] are 2-LSBMs. The structure used in [PHF86] has also connections between hidden units but no connections between observable units (see section 3.4).

The remainder of this section will introduce some more terms, which express properties of the auto-associative sets. These properties are used later to prove bounds on the number of the hidden units in terms of the size of the auto-associative set.

System state $\hat{s} = (\hat{\alpha}, \hat{\beta})$ of a Boltzmann machine represents the observable state $\hat{\alpha} \in \{0,1\}^n$ and the hidden state $\hat{\beta} \in \{0,1\}^l$. For a given machine *BM* with minimal energy E_{min}, we define the set $rep(\hat{\alpha})$ and $rep^{-1}(\hat{\beta})$ as follows:

$$rep(\hat{\alpha}) = \{\hat{\beta} \mid E(\hat{\alpha}, \hat{\beta}) = E_{min}\},$$
$$rep^{-1}(\hat{\beta}) = \{\hat{\alpha} \mid E(\hat{\alpha}, \hat{\beta}) = E_{min}\}.$$

If, for all $\hat{\alpha}$, $|rep(\hat{\alpha})| \leq 1$ holds, then we call this machine a *sharp* Boltzmann machine. Now, we can write the auto-associative set of *BM*:

$$AS(BM) = \{\hat{\alpha} \mid rep(\hat{\alpha}) \neq \emptyset\} = \bigcup_{\hat{\beta}} rep^{-1}(\hat{\beta}).$$

We say $\hat{\alpha}$ is *represented* by any vector of $rep(\hat{\alpha})$.

A *subcube* $SC(\hat{\alpha}_1, \hat{\alpha}_2)$ is a non-empty subset of $\{0,1\}^n$ defined by:

$$SC(\hat{\alpha}_1, \hat{\alpha}_2) = \{\hat{\alpha} \in \{0,1\}^n \mid \forall_i\ \alpha_i \in \{\alpha_i^1, \alpha_i^2\}\}.$$

For two vectors $\hat{\alpha}_1 = (\alpha_1^1, \ldots, \alpha_n^1)$ and $\hat{\alpha}_2 = (\alpha_1^2, \ldots, \alpha_n^2)$, we define the vector $\hat{\alpha}_1 \vee \hat{\alpha}_2$ (logical or) by:

$$\hat{\alpha}_1 \vee \hat{\alpha}_2 = (\alpha_1^1 \vee \alpha_1^2, \ldots, \alpha_n^1 \vee \alpha_n^2).$$

$\hat{\alpha}_1 \wedge \hat{\alpha}_2$ (logical and) and $\hat{\alpha}_1 \oplus \hat{\alpha}_2$ (exclusive or) are analogously defined.

Corollary 4: $SC \subseteq \{0,1\}^n$ is a subcube iff for, any $\hat{\alpha}_1, \hat{\alpha}_2 \in SC$,

$$(\hat{\alpha}_1 \vee \hat{\alpha}_2) \wedge (\hat{\gamma} \vee (\hat{\alpha}_1 \wedge \hat{\alpha}_2)) \in SC$$

holds for all $\hat{\gamma} \in \{0,1\}^n$. ♡

A *subcube decomposition* of a vector set *VS* is a set of subcubes $\{SC_1, \ldots, SC_v\}$ such that

$$VS = \bigcup_i SC_i.$$

A *minimal subcube decomposition* is a subcube decomposition with the minimum number of subcubes.

A *non-overlapping subcube decomposition* is a subcube decomposition $\{SC_1, \ldots, SC_v\}$ which satisfies:

$$i \neq j \quad \Longrightarrow \quad SC_i \cap SC_j = \emptyset.$$

A vector component t of a subcube $SC \subseteq \{0,1\}^n$ is said to be variable if, for all $\alpha \in \{0,1\}^n$, the following holds:

$$\hat{\alpha} \in SC \quad \Longleftrightarrow \quad \hat{\alpha} \oplus \hat{e}_t \in SC,$$

where $\hat{e}_t \in \{0,1\}^n$ denotes throughout this paper a unit vector:

$$\hat{e}_t = (e_1^t, \ldots, e_n^t), \qquad e_i^t = \begin{cases} 1 & \text{if } i = t, \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 4: If $AS(BM) \subset \{0,1\}^n$ for a 2-LSBM BM, then for all hidden states $\hat{\beta}$ of BM the set $rep^{-1}(\hat{\beta})$ is either a subcube or the empty set.

Proof: The lemma clearly holds if $|rep^{-1}(\hat{\beta})| \leq 1$. Therefore, assume that $\{\hat{\alpha}_1, \hat{\alpha}_2\} \subseteq rep^{-1}(\hat{\beta})$ and $\hat{\alpha}_1 \neq \hat{\alpha}_2$. It has to be shown that the vector components which are only in one of the vectors can be either 1 or 0 and the remaining vector is still in $rep^{-1}(\hat{\beta})$.

For any component t we have, for $i = 1, 2$:

$$E(\hat{\alpha}_i, \hat{\beta}) = E_{min} \quad \text{and} \quad E(\hat{\alpha}_i \oplus \hat{e}_t, \hat{\beta}) \geq E_{min}.$$

Without loss of generality let $\alpha_t^1 = 0$ and $\alpha_t^2 = 1$. Then we have:

$$E(\hat{\alpha}_2, \hat{\beta}) - E(\hat{\alpha}_2 \oplus \hat{e}_t, \hat{\beta}) = \Theta_t^o - \sum_j w_{tj}\beta_j \leq 0 \quad \text{and}$$

$$E(\hat{\alpha}_1, \hat{\beta}) - E(\hat{\alpha}_1 \oplus \hat{e}_t, \hat{\beta}) = \Theta_t^o - \sum_j w_{tj}\beta_j \geq 0.$$

Therefore, $\Theta_t^o - \sum_j w_{tj}\beta_j = 0$ holds. Hence it follows that all contribution to the energy of the vector components in which $\hat{\alpha}_1$ and $\hat{\alpha}_2$ differ must be 0 and can be ignored. This fulfills the condition of Corollary 4, which proves the assertion. ♡

Corollary 5: Each 2-LSBM BM defines a subcube decomposition D of its auto-associative set:

$$D = \{rep^{-1}(\hat{\beta}_i) \,|\, |rep^{-1}(\hat{\beta}_i)| > 0\}.$$

Proof: Each set $rep^{-1}(\hat{\beta})$ is a subcube or the empty set (Lemma 4). The definition of the auto-associative set gives $AS(BM) = \bigcup_{\hat{\beta}} rep^{-1}(\hat{\beta})$. ♡

Corollary 6: The subcube decomposition according to Corollary 5 of a sharp 2-LSBM is non-overlapping.

Proof: Assume $\hat{\alpha} \in rep^{-1}(\hat{\beta}_1) \cap rep^{-1}(\hat{\beta}_2)$ for $\hat{\beta}_1 \neq \hat{\beta}_2$. Then $|rep(\alpha)| \geq 2$, which contradicts the definition of a sharp Boltzmann machine. ♡

Remark: A description of a vector set VS with a subcube decomposition is equivalent to the expression as sum of products of the boolean function

$$f(\hat{\alpha}) = \begin{cases} 1 & \hat{\alpha} \in VS \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, it is equally hard to find a minimal subcube decomposition of a vector set and to find a minimal expression as a sum of products of a boolean function.

Example: A minimal subcube decomposition of the vector set

$$VS = \{(0, 0, x_3, x_4), (x_1', x_2', 0, 0) \,|\, x_3, x_4, x_1', x_2' \in \{0, 1\}\}$$

has two subcubes. A 2-LSBM with one hidden unit and VS as its auto-associative set can be built. The minimal non-overlapping subcube decomposition has at least three subcubes. Therefore, the minimal sharp 2-LSBM has at least two hidden units.

This example shows, that sharp Boltzmann machines actually form a distinct class, and not all machines can be transformed into a sharp version with the same auto-associative set.

3.1 Bounds on the number of hidden units of 2-LSBMs

Similarly to theorem 2, we can estimate the space needed to describe a 2-LSBM.

Corollary 7: For any 2-LSBM with n observable units and l hidden units, there exists an equivalent version, which can be described with less than $(n \cdot l)^3$ bits.

Proof: We can use the same proof as for Theorem 2, but with $m = n \cdot l + n + l$. ♡

From the previous lemma we can derive a lower bound on the number of hidden units of a 2-LSBM in terms of the size of the vectors.

Corollary 8: There exists a vector set $VS \subseteq \{0,1\}^n$ such that its implementation on a 2-LSBM needs at least $\frac{2^{n/3}}{n}$ hidden units.

Proof: There are less than $2^{(n \cdot l)^3}$ distinct 2-LSBMs with n observable units and l hidden units, but there are 2^{2^n} different subsets in $\{0,1\}^n$. We can derive

$$2^{(n \cdot l)^3} \geq 2^{2^n} \qquad \Longrightarrow \qquad l > \frac{2^{n/3}}{n}.$$ ♡

In the following, we consider the relation between the size of a subcube decomposition of a vector set VS and the number of hidden units needed to implement VS as an auto-associative set of a 2-LSBM. For the upper and lower bounds, an example will be given which exactly matches the bounds. Therefore, our bounds are tight.

Theorem 3 (lower bound): Any vector set with a minimal subcube decomposition consisting of v subcubs needs at least $\lceil log(v) \rceil$ hidden units for its implementation as an auto-associative set on a 2-LSBM.

Proof: Each 2-LSBM defines a subcube decomposition (Corollary 1). Therefore, to code all the v subcubes, we need at least $\lceil log(v) \rceil$ hidden units. ♡

Lemma 5: There exists a vector set whose minimal subcube decomposition consists of v subcubes and which can be implemented as an auto-associative set on a 2-LSBM with $log(v)$ hidden units.

Proof: The minimal subcube decomposition of the vector set $VS = \{\hat{e}_v\}$ containing all unique vectors has n subcubes, since no two vectors could be in the same subcube. A 2-LSBM can be built with $log(n)$ hidden units, which has VS as its auto-associative set. ♡

Lemma 6: If a non-overlapping subcube decomposition of VS with v subcubes is given, then a 2-LSBM with auto-associative set VS and $v-1$ hidden units can be built.

Proof: The first subcube SC_1 is represented by $\hat{\beta}_1 = \hat{0}$ ($\hat{0}$*denotesthevector*$(0,\ldots,0)$) and $\hat{\alpha}_1 \in SC_1$. Assume without loss of generality that $\hat{\alpha}_1 \neq \hat{0}$.

The thresholds for the observable units are:

$$\Theta_i^o = \begin{cases} \epsilon - 2\epsilon\alpha_i^1 & \text{if } i \text{ is not a variable component in } SC_1, \\ 0 & \text{otherwise.} \end{cases}$$

where ϵ is given by $\sum_i \Theta_i^o \alpha_i^1 = -1$.

All other subcubes are represented by a unique vector. For $j = 2,\ldots,v$, SC_j is represented by $\hat{\beta}_j = \hat{e}_{j-1} \in \{0,1\}^{v-1}$ and $\hat{\alpha}_j \in SC_j$, and

$$w_{i,j-1} = \begin{cases} 4\alpha_i^j + \Theta_i^o - 2 & \text{if } i \text{ is not a variable component in } SC_j, \\ 0 & \text{otherwise.} \end{cases}$$

The thresholds for the hidden units are:

$$\Theta_{j-1}^h = -1 - \sum_i (\Theta_i^o - w_{i,j-1})\alpha_i^j.$$

♡

Theorem 4 (upper bound): For all given vector sets with v elements, there exists a 2-LSBM with at most $v-1$ hidden units.

Proof: A non-overlapping subcube decomposition of a vector set is given by putting only one vector in each subcube. According to Lemma 6, a 2-LSBM with $v-1$ hidden units can be built. ♡

Lemma 7: There exists a vector set VS with v elements such that all 2-LSBMs that have VS as their auto-associative set have at least $v-1$ hidden units.

To prove the above lemma, we first show the following lemma:

Lemma 8: Let $AS \subseteq \{0,1\}^n$ be the auto-associative set of a 2-LSBM. Let $\{(\hat{\alpha}_i, \hat{\beta}_i) \mid \hat{\alpha}_i \in rep^{-1}(\hat{\beta}_i) \subseteq AS\}$ be a subset of system states and $\{a_i\}$ denote a sequence of real values for which $\hat{0} = \sum_i a_i \hat{\beta}_i$ and $0 = \sum_i a_i$ hold. If there exists a t such that

$$\forall\, i,j \;:\quad (-1)^{\alpha_t^i}(-1)^{\alpha_t^j} a_i a_j \geq 0$$

, then, for any $a_t \neq 0$, $\hat{\alpha}_i \oplus \hat{e}_t \in rep^{-1}(\hat{\beta}_i) \subseteq AS$ holds.

Proof: Without loss of generality, we assume that

$$\forall\, i \;:\quad \alpha_t^i = \begin{cases} 1 & a_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad \Longrightarrow \quad \forall_i \;:\quad a_i \hat{e}_t^T(\hat{\Theta}_o - W\hat{\beta}_i) \leq 0.$$

It follows from this assumption that:

$$\sum_i a_i \hat{e}_t^T(\hat{\Theta}_o - W\hat{\beta}_i) = \hat{e}_t^T(\hat{\Theta}_o \cdot \sum_i a_i - W\sum_i a_i\hat{\beta}_i) = \hat{e}_t^T(\hat{\Theta}_o \cdot 0 - W \cdot \hat{0}) = 0$$

$$\Longrightarrow \quad \forall_i \;:\quad a_i \hat{e}_t^T(\hat{\Theta}_o - W\hat{\beta}_i) = 0.$$

Therefore, for any $a_i \neq 0$, $\alpha_i \oplus e_t \in rep^{-1}(\beta_i)$ holds. ♡

Example: The vector set AS_{xor} (exclusive or, see section 1.3) cannot be implemented on any 2-LSBM with 2-hidden units. The minimal subcube decomposition has 4 subcubes and any representation of these subcubes on the two hidden units leads to a contradiction with Lemma 8.

According to Lemma 8, we can construct a vector set which proves the assertion of lemma 7.

Proof of Lemma 7: We define the vector set $VS = \{\hat{\alpha}_1, \ldots, \alpha_v\}$ with the following property:

$$\forall\, \hat{\gamma} \in \{0,1\}^v \quad \exists\, t \quad \forall\, i : \quad \alpha_i^t = \gamma_i.$$

Each vector in VS has 2^v components and

$$\hat{\alpha} \in VS \quad \Longrightarrow \quad \hat{\alpha} \oplus \hat{e}_j \notin VS.$$

Further, the vectors of VS are represented by $\hat{\beta}_i \in rep(\hat{\alpha}_i)$. If we assume that the 2-LSBM has less than $v-1$ hidden units, then

$$\exists\, \{a_i\} : \quad \sum_i a_i \hat{\beta}_i = \hat{0}, \quad \sum_i a_i = 0, \quad \text{and} \quad \sum_i (a_i)^2 \neq 0.$$

Because of the given construction, we have

$$\exists\, t : \quad \alpha_i^t = \begin{cases} 1 & a_i > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, all conditions for Lemma 8 are fulfilled and there is at least one $a_i \neq 0$ that leads to a contradiction. ♡

3.2 Lower bound for one vector sets

The bounds in section 3.1 are both tight, but not readily applicable to real problems. The examples given in the proofs are both small subsets of huge vector spaces. Let us consider the vector sets containing the whole vector space but one vector. From among them, we choose $VS = \{0,1\}^n / \{\hat{0}\}$. The other sets can be derived using Lemma 1.

Lemma 9: A 2-LSBM which defines the following minimal and non-overlapping subcube decomposition of $AS = \{0,1\}^n / \{\hat{0}\}$ must have at least $n-1$ hidden units. The subcube decomposition is given by the following n subcubes:

$$SC_i = \{(\alpha_1, \ldots, \alpha_n) \,|\, \forall_{j<i} \; : \; \alpha_j = 0 \wedge \alpha_i = 1\}.$$

Proof: If the machine consists of less than $n-1$ hidden units, then there is at least one combination of representing hidden states which fulfills the following equations:

$$\hat{0} = \sum_i a_i \hat{\beta}_i, \qquad 0 = \sum_i a_i, \quad \text{and} \quad 0 \neq \sum_i (a_i)^2.$$

We choose $t = max\{i \mid a_i \neq 0\}$ and from each subcube SC_j, $j \leq t$,

$$\hat{\alpha}_j = \begin{cases} \hat{e}_j & a_j a_t \leq 0 \\ \hat{e}_j \vee \hat{e}_t & \text{otherwise.} \end{cases}$$

This set of system states $\{(\hat{\alpha}_1, \hat{\beta}_1), \ldots, (\hat{\alpha}_t, \hat{\beta}_t)\}$ fulfills the conditions of Lemma 8 at component t, a contradiction to $\hat{e}_t \oplus \hat{e}_t = \hat{0} \in VS$. ♡

The above lemma shows a lower bound only for one particular subcube decomposition. However, we expect the following result:

Conjecture: There is no (sharp) 2-LSBM with less than $n-1$ hidden units and auto-associative set $\{0,1\}^n/\{\hat{0}\}$.

3.3 Implementation of k-variable boolean functions

A k-variable boolean function $f : \{0,1\}^k \rightarrow \{0,1\}$ can be represented as a vector set

$$VS = \{(x_1, \ldots, x_k, f(x_1, \ldots, x_k))\}.$$

VS contains 2^k vectors. If VS is the auto-associative set of a Boltzmann machine BM, then the function is said to be *implemented* by BM.

We define the *representation number* L_f as the minimal number of hidden units that are needed to implement the function f on a 2-LSBM.

We define the *maximal representation number* $L : \mathbb{N} \rightarrow \mathbb{N}$ as follows.

$$L(k) = max\,\{L_f \mid f \text{ is a k-variable boolean function}\}.$$

With Lemma 1 we can obtain from a Boltzmann machine, which implements a function, Boltzmann machines for the inverse function or any function with inverse inputs. Therefore, a function which can be realized by negation or permutation of the inputs of another function f has the same representation number as f.

Experimental Results: $L(k)$ can be calculated for small k by constructing a Boltzmann machine for each function f with k variables. But, with increasing k, not only does the number of functions increase exponentially but also the number of possible representations on the hidden units increases even more strongly. For any given k, there exist 2^{2^k} functions and, for each function, there are $2^{L(k)*2^k}$ representations. Even if we consider only functions that cannot be realized by negation or permutation of inputs, we still have to consider at least $2^{(2^k-k^2)}$ functions.

$L(1) = 1$. For $k = 1$, there are 4 functions with 4 representations.

$L(2) = 3$. For $k = 2$, there are 16 functions with 2^{12} representations. Only the exclusive-or function and its negation need more than 2 hidden units.

$L(3) = 4$. For $k = 3$, there are 256 functions with 2^{32} representations. Only eight functions need more than three hidden units. All can be realized by negation or permutation of

$$f_3(x_1, x_2, x_3) = \begin{cases} 1 & x_1 = x_2 = x_3, \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 10: The representational number L_f of a k-variable boolean function f whose canonical form has t products is bounded by:

$$L_f < t \cdot k.$$

Proof: There is always a non-overlapping subcube decomposition with $t \cdot k$ subcubes. Theorem 4 gives a bound. ♡

Lemma 11: The representational number $L_{f_k^0}$ of the k-variable boolean function $f(\hat{x}) = \overline{x_1} \wedge \cdots \wedge \overline{x_k}$ has the following lower bound:

$$L_{f_k^0} \geq log(k+1).$$

Proof: Any subcube decomposition of its auto-associative set has at least $k+1$ subcubes. Theorem 3 gives a bound. ♡

Theorem 5: The maximal representation number $L(k)$ has the following bounds:

$$\frac{2^{k+1/3}}{k+1} < L(k) < 2^k.$$

The lower bound is given by Corollary 8. The upper bound is given by Theorem 4. ♡

Remark: We have not found any function for which we could prove that it required more than $O(k)$ hidden units for an implementation on a 2-LSBM. One such function is the parity function par_k, defined by:

$$par_k(\hat{x}) = \begin{cases} 1 & \sum_i x_i = odd, \\ 0, & \text{otherwise.} \end{cases}$$

Any minimal subcube decomposition of the vector set $\{\hat{x}, par(\hat{x})\}$ has 2^k subcubes. But we could not find any upper bound on L_{par_k} smaller than $o(2^k)$, nor any lower bound bigger than $O(k)$. If we are not restricted to 2-LSBMs, the parity function can be implemented on a Boltzmann machine by using composition with the machine shown in Figure 1. This implementation requires less than $2k$ hidden units.

3.4 k-LSBMs and FOBMs

The results in this section are, for simplicity, derived only for 2-LSBMs. The 2-LSBM can be extended to a k-LSBM by adding more hidden layers with connections between the i-th and the $(i+1)$-st layers. The layer of the observable units is exclusively connected with the first hidden layer. This definition can be adapted to the the commonly used k-layer structure with the inputs at the first layer and the outputs at the k-th layer. If we want to compare a k-LSBM BM_1 with such a machine BM_2 (which has the same layer structure but the units representing the system response are connected to the k-th layer), then we can construct BM_1' from BM_1 by adding one unit per output unit in each hidden layer. The new units can form an information channel to transfer the information on the k-th layer to the output units. Computationally, BM_1' is at least as powerful as BM_2. A Boltzmann machine without connections among observable units is called *free observable Boltzmann machine* (FOBM), and this class includes all k-LSBMs. With some minor changes in the notation, all results also hold for FOBMs, except for Corollary 7, 8, and Theorem 5. Even more strongly, the lower bound in Theorem 4 gives only the number of hidden units which are connected to the observable units. For FOBMs we have to use Theorem 2 and Corollary 3 instead of Corollaries 7 and 8. Theorem 5 has to be replaced by:

Corollary 9: The number, l_m, of hidden units required to implement any k-variable boolean function on a FOBM is bounded by

$$\frac{2^{(k+1)/5}}{k+1} < l_m < 2^k.$$

4. Conclusion and open questions

This paper has addressed the question of determining the number of hidden units necessary for storing a vector set on a Boltzmann machine. We introduced the auto-associative set of a Boltzmann machine as the set of vectors which is memorized by the machine. Using basic operations such as bit-wise negation, intersection, and composition, related machines can be constructed from the original one. With composition we compared the computational power of switching networks and Boltzmann machines. Using the auto-associative set we derived some bounds on the number of required hidden units. One result is based on the assertion that any Boltzmann machine with a polynomial number of units can be described in polynomial space. Since there are hyperexponential number of subsets in $\{0,1\}^n$, there exist vector sets whose implementation on Boltzmann machines requires exponential number of hidden units. Further results are restricted to the class of Boltzmann machines without connections among their observable units. These results are given in terms of the size of the vector set or the size of its subcube decomposition and provide a logarithmic lower bound and a linear upper bound. Both these bounds were shown to be tight. In the last section, we applied our results to boolean functions and showed that, to implement even simplest functions, a Boltzmann machine needs at least a logarithmic number of hidden units.

Our results give the same bounds for 2-LSBMs, the simplest structure with hidden units, as for FOBMs, Boltzmann machines without connections among observable units. We would like to know if any extension of the 2-LSBM brings some advantage in terms of the required number of hidden units. More formally:

- Does there exist a vector set, such that its implementation as an auto-associative set of a 2-LSBM requires more hidden units than as an auto-associative set of a k-LSBM/FOBM ?
- If so, does there exist, for any k, a vector set such that its implementation on a k-LSBM requires more hidden units than on a $(k+1)$-LSBM ?

References

[AaK89] Aarts, E. and Korts, J. (1989). *Simulated Annealing and Boltzmann Machines.* Chichester: John Wiley and Sons

[AbJ85] Abu-Mostafa, Y. and St. Jacques, J.M. (1985). Information capacity of the Hopfield model. *IEEE Trans. on Information Theory* **31**, 461-464.

[AHS85] Ackley, D.H., Hinton, G.E., and Sejnowski, T.J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science* **9**, 147-169.

[BiY86] Binder, K. and Young, A.P. (1986). Spin glasses: Experimental facts, theoretical concepts, and open questions. *Reviews of Modern Physics* **58**, 801-976.

[Mos90] Moser, E. (1990). About learning in Boltzmann machines. Unpublished.

[PHF86] Prager, R.W., Harrison, T.D., and Fallside, F. (1986). Boltzmann machines for speech recognition. *Computer Speech and Language* **1**, 3-27.

[PaS89] Parberry, I. and Schnitger, G. (1989). Relating Boltzmann machines to conventional models of computation. *Neural Networks* **2**, 59-67.

Centre for Systems Science

• Simon Fraser University established the **Centre for Systems Science** as a provincial centre for advanced study. The objective of CSS is twofold: coordinate and combine advanced systems research at SFU, and encourage application of research results within industry in order to further the technological development of British Columbia.

• CSS has assembled three research groups to streamline its efforts. The orientation of these groups is as follows: intelligent systems, microelectronics, and computer and communications systems. Because of shared research interests, this latter group merges with LCCR (**Laboratory for Computer and Communications Research**) for the publication of technical reports.

• CSS's three groups involve over 80 skilled faculty researchers principally from SFU's Schools of Engineering Science and Computing Science, but also draw substantially on Mathematics, Physics, Geography, Philosophy, Linguistics, Psychology, Communication, and Kinesiology.